**CStorage**: Distributed Data Storage in Wireless Sensor Networks Employing Compressive Sensing

Ali Talari and Nazanin Rahnavard  
Oklahoma State University, Stillwater, OK 74078,  
Emails: {ali.talari, nazanin.rahnavard}@okstate.edu

**Abstract**—In this paper, we propose CStorage a fully distributed and efficient data storage scheme for wireless sensor networks (WSNs) based on compressive sensing (CS) techniques. CStorage requires much smaller number of transmissions compared to the existing algorithms by exploiting the compressibility of the natural signals along with the broadcast property of wireless channels. In CStorage, after a probabilistic readings dissemination phase, each node obtains one compressed sample (measurement) of the network’s readings, which are later queried in part by a data collector to recover all readings. We find the optimal parameters of CStorage and show that it considerably decreases the total number of required transmissions for distributed data storage.

**I. INTRODUCTION**

To increase the data persistence in wireless sensor networks (WSNs), it has been proposed to disseminate sensors readings throughout the network such that a data collector can query any small subset of nodes to obtain all sensors’ readings [1, 2]. However, existing algorithms either ignore the correlation and compressibility of the readings, require routing tables and cannot be distributively implemented, or incur a large number of transmissions.

It has been shown that signals collected in WSNs from natural phenomena are highly compressible due to their strong spacial correlation [3–5]. Therefore, such compressible signals including N readings from N network nodes can be reconstructed from only M ≪ N compressed samples (measurements) of the signal employing compressive sensing (CS) techniques [6, 7]. On the other hand, wireless channels have the inherent broadcast property, hence probabilistic broadcasting (PBcast) [8, 9] can be effectively utilized to disseminate nodes’ readings in the networks with minimal transmissions.

Utilizing the two aforementioned properties, we propose an efficient and flexible cross-layer data storage algorithm referred to as compressive sensing data storage (CStorage), which considerably reduces the number of required transmissions for distributed data storage in WSNs. In CStorage, when the data dissemination phase using PBcast is finished a data collector can query M ≪ N measurements from any set of M nodes and recover all N readings. CStorage is fully scalable and distributed since nodes independently make decisions without employing any routing table.

The paper is organized as follows. Section II provides a brief review on compressive sensing, PBcast, and the related work. In Section III, we propose CStorage. In Section IV we evaluate the performance of CStorage and compare it to existing algorithms. Finally, Section V concludes the paper.

**II. BACKGROUND AND RELATED WORK**

In this section, we provide the necessary background to design CStorage.

**A. Compressive Sensing in WSNs**

Let us consider a WSN with N nodes collecting a natural signal \( \mathbf{x} = [x_1, x_2, \ldots, x_N]^T \), where \( x_i \) represents the reading of the \( i^{th} \) sensor. Due to spacial correlation of sensor readings \( \mathbf{x} \) may be represented by a K-sparse signal \( \mathbf{\theta} \) with only \( K \ll N \) large coefficients in some appropriate transform basis \( \Psi \), where \( \mathbf{x} = \Psi \mathbf{\theta} \) [4, 5, 10]. More precisely, when the coefficients of \( \mathbf{\theta} \) are sorted based on their absolute magnitude, they decay faster than \( C i^{-\beta} \) for \( 0 < \beta \leq 1 \) and a constant \( C \) [4, 5, 10].

Consider a natural signal \( \mathbf{x} \) collected from a WSN that is \( K \)-sparse in some proper basis \( \Psi \). CS techniques are able to recover \( \mathbf{x} \) from only \( M \ll N \) random projections (also called measurements or compressed samples) of \( \mathbf{x} \), where \( M \geq O(K \log N) \) [6, 7]. Generally, CS is composed of two following key components.

**Signal Sampling**: The random projections are generated by \( \mathbf{y} = \Phi \mathbf{x} \), where \( \Phi \) is a well-chosen \( M \times N \) random matrix, called projection matrix.

**Signal Recovery**: Signal reconstruction can be done by finding the estimate \( \hat{\mathbf{\theta}} \) (and accordingly \( \hat{\mathbf{x}} = \Psi \hat{\mathbf{\theta}} \)) from the system of linear equations \( \mathbf{y} = \Phi \Psi \mathbf{\theta} \). This is an underdetermined system with infinitely many solutions. However, the knowledge of \( \mathbf{\theta} \) being a sparse signal allows us to have a successful reconstruction w.h.p. It is shown that \( \hat{\mathbf{\theta}} \) can be estimated via solving the \( \ell_1 \) optimization problem given by [6, 7, 11, 12]

\[
\hat{\mathbf{\theta}} = \arg\min \|\mathbf{\theta}\|_1, \text{ s.t. } \mathbf{y} = \Phi \Psi \mathbf{\theta},
\]

where \( \|\mathbf{\theta}\|_1 = \sum_{i=1}^{N} |\theta_i| \). The \( \ell_1 \) optimization problem (1) can be solved with linear programming techniques called basis pursuit (BP) [12]. Later, we employ CS in CStorage design.

**B. Probabilistic Broadcasting: PBcast**

Consider a WSN with \( N \) nodes randomly deployed over a field of size \( A = 1 \times 1 \). To ensure the connectivity of the network we assume all nodes have identical transmission range
of $r_t^2 > \frac{4 \ln(N)}{\pi N}$ [13], and two nodes can communicate if their Euclidian distance is less than $r_t$.

In PBcast, a node $n_i$ broadcasts its reading $x_i$ instead of routing it to a specific neighbor. Therefore, all neighbors of the $n_i$ receive $x_i$ and would be able to store it (in CStorage a compressed measurements is stored instead of readings as will be discussed later). Each node that has received $x_i$ for the first time will rebroadcast it with probability $p$ and this probabilistic forwarding continues. Figure 1 shows the average fraction of network nodes $R(p)$ that receive a particular reading after the PBcast and the total number of transmissions $N_t(p)$ versus the forwarding probability $p$ for a connected WSN with $N = 10^4$ and $r_t = 0.02568$.

Fig. 1. The fraction of nodes receiving a reading $R$ and the number of transmissions $N_t$ versus forwarding probability $p$ for PBcast.

As we see in Figure 1, at $p \approx 0.24$ a large fraction of nodes receive the broadcast of reading $x_i$. Moreover, we can observe that although increasing $p$ beyond $p \approx 0.24$ does not significantly contribute to the delivery of $x_i$, it considerably increases the number of transmissions (almost linear in $p$). Therefore a well-chosen small forwarding probability $p^*=0.24$ would be sufficient to ensure a large fraction of nodes in a network have received a reading.

It has been shown that $p^*$ is close to the probability $p_c^G$ that a giant component appears in the network, where asymptotically $p_c^G \approx \frac{1.44}{N r_t^2}$ [8, 9]. For our given network topology with $N = 10^4$ and $r_t = 0.02568$ we have $p_c^G = 0.23$. Therefore, $p^*$ can be approximated with $p_c^G$ when $N$ is large enough.

C. Related Work

Authors in [14] and [15] have proposed to employ gossiping and random walks, respectively, to disseminated reading in a large-scale WSN and form measurements at nodes. As we later show, CStorage outperforms these algorithms in the of measurements due to efficient utilization of PBcast.

Authors in [2, 16, 17] have proposed data storage algorithms for sensor networks based on error correction codes. Although these algorithms are efficiently designed, they have not exploit the compressibility of the signals in a sensor network to reduce the number of transmissions.

Finally, authors in [3–5, 10, 18] have considered the compressibility of the data collected. However, they have assumed that either routing tables are available or the measurements are collected at a central sink. Therefore, these schemes cannot be implemented in a distributed large-scale WSN.

III. PROPOSED ALGORITHM: CStorage

In this section, we propose and discuss CStorage.

A. CStorage

In CStorage, node $n_j, j \in \{1, 2, \ldots, N\}$, maintains a CS measurement $y_{jj}$, where $y_{jj}$ is formed as $y_{jj} = \phi_j x_j$ and $\phi_j$ is an $N$-dimensional row vector. Let $\Phi_{tot} = [\phi_1^T \phi_2^T \ldots \phi_N^T]^T$ and $y_{tot} = [y_{11} y_{22} \ldots y_{NN}]^T$. Further, let $\varphi_{j,i}$ be the element at the $j$th row and the $i$th column of $\Phi_{tot}$. CStorage steps are as follows:

1) $N_s \geq M$ nodes randomly select themselves as a source node and broadcast their readings to their neighbors.
2) Upon the reception of reading $i$ ($x_i$) for the first time, node $l$ performs the following:
   a) chooses $\varphi_{l,i} = \pm 1$ with equal probabilities and adds $\varphi_{l,i} x_i$ to $y_l$.
   b) Broadcasts $x_i$ with probability $p$ (PBcast).

When the transmissions are finished, a $\Phi_{tot}$ with approximately $N_s$ non-zero entries per row (for a large enough $p$) and $y_{tot} = \Phi_{tot} x$ have been distributed in the network nodes. Therefore, a data collector may gather $M$ measurements $y \in \mathbb{R}^M$ and the corresponding $\phi_j$’s from an arbitrary set of $M$ nodes and obtain the measurement matrix $\Phi_c \in \mathbb{R}^{M \times N}$ (by putting together the $M$ collected rows of $\Phi_{tot}$). Next, the data collector obtains $\hat{x}$ an estimate of $\mathbb{E}$ employing BP via solving (1).

Note that in CStorage the only information the network nodes need to know is the value of $N_s$ (or equivalently the probability $\frac{N_s}{N}$ where a node selects itself as a source node) and the value of $p$, which may be preprogrammed into nodes before the network deployment.

B. Forming $\Phi_{tot}$ Employing CStorage

Let us consider a small network with $N = 5$ nodes as shown in Figure 2 and investigate one PBcast. Clearly, at the beginning, $\Phi_{tot}$ is an all-zero $5 \times 5$ matrix. Assume that node $n_1$ broadcasts its reading $x_1$ (see Figure 2). Since $n_2$ and $n_3$ are in the transmission range of $n_1$, they would receive $x_1$. Node $n_2$ multiplies $x_1$ by $\varphi_{2,1}$ and adds $\varphi_{2,1} x_1$ to $y_2$. Similarly, $n_3$ multiplies $x_1$ with $\varphi_{3,1}$ and adds $\varphi_{3,1} x_1$ to $y_3$.

The resulting $\Phi_{tot}$ matrix at this step is given by (2).

$$\Phi_{tot} = \begin{pmatrix} \varphi_{1,1} & 0 & 0 & 0 & 0 \\ \varphi_{2,1} & 0 & 0 & 0 & 0 \\ \varphi_{3,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$ (2)

At this point, $n_2$ and $n_3$ independently decide whether to broadcast $x_1$ with probability $p$ or not. Assume that $n_2$ decides to broadcast $x_1$. Node $n_4$ would receive $x_1$ and adds $\varphi_{4,1} x_1$ to $y_4$. However, we assume that $n_3$ and $n_4$ decide not to rebroadcast $x_1$. Thus, the PBcast of $x_1$ is over and the matrix $\Phi_{tot}$ obtains the form of (3). Note that in Figure 2, we have shown the transmitting nodes with a dark color, while the rest of the nodes are shown by white color.
uniformly at random. Therefore, for \( p \approx p^* \approx p^G \) the resulting \( \Phi_{tot} \) and \( \Phi_c \) from CStorage have approximately \( N_s \) non-zero entries placed uniformly at random in each row. Further, generating a dense \( \Phi_c \) with \( N \) non-zero entries per row imposes a large communication load, i.e., \( N_t(p) \times N \) transmissions. As a result, generating a sparse \( \Phi_c \) is of interest since the number of transmissions would significantly reduce.

C. Selection of Transform Basis \( \Psi \)

In [11, 19] authors showed that for any dense orthonormal \( \Psi \), e.g., Fourier transform basis, the number of required measurements for the recovery of a \( K \)-sparse signal can be obtained from \( M \geq C'K\log^4 N \), where \( C' \) is a constant. This implies that if the signal is compressible in some dense orthonormal basis we may employ a sparse \( \Phi \) with at least one non-zero placed independently and randomly per row and satisfy CS conditions [11]. Therefore, we may also employ a sparse \( \Phi_c \) to decrease the number of transmissions in CStorage without deteriorating the CS reconstruction performance as long as the rows of \( \Phi_c \) are randomly selected, or equivalently are linearly independent.

The basis \( \Psi \), under which the signal is sparse or compressible, depends on the nature of the physical phenomenon. For example, in [5] it was shown that the temperature readings of a sensor network are compressible in discrete cosine transform (DCT) basis. Therefore, without loss of generality in the rest of our paper we assume that \( \Psi \) is the DCT transform basis (while it also may be any other dense orthonormal basis).

D. Optimum Values of \( N_s \) and \( p \)

On one hand, \( p \) and \( N_s \) affect the properties of the \( \Phi_{tot} \) generated in the network, and on the other hand, they directly determine the number of transmissions performed in CStorage, i.e., \( n_{tot} = N_t(p) \times N_s \). Hence, we need to find the optimum values of \( p \) and \( N_s \) for which the minimum \( n_{tot} \) is obtained while \( \Phi_c \) matrix has \( M \) independent rows.

Assume that CStorage has been performed with parameters \( N_s \) and \( p \). Clearly, we would have non-zero entries only in the columns of \( \Phi_c \) that correspond to \( N_s \) source nodes that broadcast their own reading. Further, the rest of \( N - N_s \) columns are all zero and may not contribute in forming independent rows for \( \Phi_c \). As shown in Figure 3, consider the submatrix \( \Phi_s \in \mathbb{R}^{M \times N_s} \) of \( \Phi_{tot} \) formed by selecting the \( N_s \) columns of \( \Phi_{tot} \) corresponding to the \( N_s \) nodes broadcasting their readings, and \( M \) rows corresponding to \( M \) measurements obtained by data collector. Clearly, if \( \Phi_s \) has rank at least \( M \), then the rows of \( \Phi_s \) and consequently \( \Phi_c \) are independent.

\[
\Phi_{tot} = \begin{pmatrix}
\varphi_{1,1} & 0 & 0 & 0 & 0 \\
\varphi_{2,1} & 0 & 0 & 0 & 0 \\
\varphi_{3,1} & 0 & 0 & 0 & 0 \\
\varphi_{4,1} & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]  \hspace{1cm} (3)

\[
\Phi_{tot} \in \mathbb{R}^{N \times N} \\
\Phi_s \in \mathbb{R}^{M \times N_s}
\]

Fig. 2. Network with \( N = 5 \) and \( n_1 \) transmitting \( x_1 \) employing PBcast.

The same procedure is performed for \( N_s \) source nodes selected uniformly at random. We formulate the rank of \( \Phi_s \) as a function of the ratio of nodes receiving a particular transmission \( R(p) \) (see Figure 1) and \( N_t \) in the following theorem.

**Theorem 1:** Let \( \Phi_{tot} \) be the measurement matrix generated one row per node employing CStorage. Further, let \( \Phi_s \) be the submatrix from \( \Phi_{tot} \) by selecting any desired \( M \) rows and \( N_s \) columns corresponding to \( N_s \) source nodes. \( r(\cdot) \) the expected rank of the matrix \( \Phi_s \), after \( j \)th transmission out of \( N_s \) transmissions of CStorage is given by the following recursive equation:

\[
\begin{align*}
    r(0) &= 0, \\
    r(j) &= 1 - (1 - R(p))^{M-r(j-1)} + r(j-1), j \in \{1, 2, \ldots, N_s\}.
\end{align*}
\]  \hspace{1cm} (4)

**Proof:**

Clearly, if the network nodes uniformly receive a reading PBcast then \( R(p) \) would also be the probability that each node receives the reading. However, generally the dissemination is not uniform, e.g., nodes on the border of network would receive fewer readings. Let \( R_{N_t}(p) \) denote the probability that a node receives all \( N_s \) transmissions. Clearly, in a uniform distribution \( R_{N_t}(p) = R(p)^{N_s} \). However, authors in [8, 9] found the bounds on \( R_{N_t}(p) \) in a network where a ratio of \( R(p) \) nodes non-uniformly receive a PBcast as follow:

\[
R(p)^{N_s} \leq R_{N_t}(p) \leq R(p).
\]

It is worth noting that the bounds become tighter as \( N \to \infty \). Therefore, we consider the worst case and assume that all nodes of the network uniformly receive each PBcast with probability \( R(p) \).

Let \( t(j) \) denote the probability that at least one independent row is added to \( \Phi_s \) after \( j \)th PBcast. Further, let \( r(j) \) be the expected value of the rank of \( \Phi_s \). At the beginning, \( \Phi_s \) is an all-zero matrix; hence it has rank 0, i.e., \( r(0) = 0 \). When the first broadcast is performed, if at least one node out of \( M \) nodes of interest receives this broadcast the rank of the matrix \( \Phi_s \) increases to 1. There is also a possibility that none of these nodes receive this broadcast. For the first transmission, we would have \( t(1) = 1 - (1 - R(p))^M \). Therefore, regardless
of the value assigned to all \( \varphi_{i,j} \)'s in the first transmission, one independent row is added to \( \Phi_s \) by this transmission with probability \( t(1) \). Hence the expected number of independent rows of \( \Phi_s \) becomes \( r(1) = 1 \times t(1) \).

If at least one node out of \( M \) nodes of interest has received the first transmission, the next broadcast should be received by one node out of \( M - 1 \) nodes so that the rank of \( \Phi_s \) increases to \( 2 \). Therefore, we need to receive the second transmission by \( M - 1 \) nodes given that the first transmission was successful. Consequently, the second transmission should be received by at least one of the \( M - r(1) \) nodes in expectation so that a new independent row can be added to \( \Phi_s \).

However, a new independent row would not be added to \( \Phi_s \) if and only if the following event occurs. Assume an arbitrary set of nodes have received the first transmission. In addition, assume exactly the same set of nodes have received the second transmission. Further, assume that all these nodes have selected the same random \( \varphi_{i,j} \) in the second reception as they selected in the first reception. In this case, all the rows of \( \Phi_s \) with non-zero entries would look alike, hence the rank remains \( 1 \). However, it is not hard to show that such a rare event happens with probability

\[
\sum_{l=1}^{N} \left( \frac{N}{l} \right) R(p)^l (1 - R(p))^{N-l} \frac{2}{\left( \frac{N}{l} \right)^2} - 1,
\]

which is almost equal to zero for practical values of \( N \).

Therefore, a new independent row is indeed added to \( \Phi_s \) if one of the \( M - r(1) \) nodes in expectation receive the new transmission. Therefore, we have \( t(2) = 1 - (1 - R(p))^{M - r(1)} \) and \( r(2) = r(1) + 1 \times t(2) \). Similarly, \( r(j) \) can be found recursively as \( r(j) = r(j-1) + 1 \times t(j) \) with \( t(j) = 1 - (1 - R(p))^{M - r(j-1)} \) as given in the Theorem 1.

As an example, we set \( N = 10^4 \) and \( M = 700 \) and employ Theorem 1 along with the values of \( R(p) \) given in Figure 1 to find the rank of \( \Phi_s \) versus \( p \) and \( N_s \) as shown in Figure 4.

Based on the results observed in Figure 4, we find the total number of transmissions \( n_{tot} = N_s \times N_t(p) \) for the values of \( p \) and \( N_s \) for which \( \frac{r(N_s)}{M} \geq 0.9999 \) versus \( N_s - M + 1 \) in Figure 5 (we have plotted \( n_{tot} \) versus \( N_s - M + 1 \) to have a better view on the values close to \( N_s = M \) on a log-scale axis). Note that we have not fixed \( p \) to obtain the curve in Figure 4, but rather relaxed the value of \( p \) and searched for the minimum \( n_{tot} \).

![Figure 4](image-url)

**Figure 4.** \( \frac{r(N_s)}{M} \) versus \( p \) and \( N_s \).

Figure 4 shows that for \( N_s \geq M \) the rank of matrix \( \Phi_s \) approaches to \( M \) for a large enough \( p \). More importantly, it shows that as \( N_s \) increases a suitable matrix can be generate with a smaller value of \( p \). Consequently, we see an interesting trade-off since increasing \( N_s \) linearly increases the total number of transmissions, i.e., \( N_s \times N_t(p) \), while it non-linearly reduces the required \( p \) and consequently \( N_t(p) \) and \( n_{tot} \).

![Figure 5](image-url)

**Figure 5.** The total number of transmissions \( n_{tot} \) required to generate a suitable \( \Phi_s \) and \( \Phi_c \) with \( \frac{r(N_s)}{M} \geq 0.9999 \) versus \( N_s - M + 1 \).

The aforementioned trade-off between \( N_s \) and \( p \) with \( n_{tot} \) can be observed in Figure 5, and we can see that the number of transmissions is minimized when \( N_s \) is slightly larger than \( M \). Figure 5 shows that \( n_{tot} \) is minimized for \( N_s = 702 \), which is obtained for \( p = 0.24 = p^* \).

**IV. PERFORMANCE EVALUATION OF CSTORAGE**

We generate a compressible signal with \( \beta = \frac{7}{8} \) similar to [4] and obtain a compressible signal by \( x = \Psi \hat{x} \). We run our simulation for a randomly deployed WSN as described in Section II-B with \( N = 10^4 \), \( M = 700 \), and \( r_1 = 0.02568 \). Figures 6(a) and 6(b) show the normalized reconstruction error \( \| \hat{x} - x \|_2^2 \) and \( n_{tot} \) employing CSTorage versus \( p \) for various \( N_s \)’s, where \( \hat{x} \) is the reconstructed estimate of \( x \) at data collector.

Figure 6(a) confirms that the minimum required \( p \) is about \( p^* = 0.24 \). Further, we can see that for \( N_s \) slightly larger than \( M = 700 \) the reconstruction of \( x \) is almost as good as the ideal case with a dense \( \Phi_c \). Therefore, increasing \( N_s \) further does not reduce the reconstruction error while it considerably increases the total number of transmissions \( n_{tot} \) (Figure 6(b)). Moreover, we can see that due to the correlation of data in our network with only \( M \approx 0.07N \) measurements all \( N \) readings have been reconstructed.

Further, we compare the total number of transmissions \( n_{tot} \) employing the same WSN employing various data storage algorithms that do not need routing tables in Table I. Note that in the last row of Table I we have evaluated the number of required transmissions if CSTorage was implemented utilizing random walks as employed in [2, 15, 17] instead of PBcast. Note that the signal reconstruction quality in all algorithms is equal. Table I shows that we have decreased the number of
transmissions more than one order of magnitude compared to existing protocols. This is an excellent reduction in the number of transmissions that results in energy saving to a great extent and enhances the lifetime of the WSN while it does not add any more memory requirement or computational complexity to sensor nodes.

Parameters of CStorage and showed that \( N_s \) needs to be slightly large than \( M \) to achieve the minimum number of required transmissions, which is considerably lower than the required number of transmissions in existing protocols for distributed data storage in WSNs. In the future work, CStorage will be extended considering the spatio-temporal correlation of the readings.

**References**


